

Project Report
on
Predicting Life Expectancy
using
Machine Learning
By
Idrees Tyre Wala

Predicting Life Expectancy Using Machine Learning

1.	Introduction.....	4
	Overview	4
	Purpose.....	5
2.	Literature Survey	6
	Existing Problem.....	6
	Proposed Solution.....	6
3.	Theoretical Analysis	7
	Block Diagram.....	7
	Hardware / Software Designing	8
4.	Experimental Investigations.	9
5.	Flowchart	16
6.	Result	18
7.	Advantages & Disadvantages	20
8.	Applications.....	21
9.	Conclusion	22
10.	Future Scope.....	23
11.	Bibliography	24

1. Introduction

Since ancient times, there are a lot of change in the behaviours and cultures of people in different places. According to their way of living, the health care and life expectancy of people varies among each other. These differences are may be based on various factors such as Regional variations, Economic Circumstances, Sex Differences, Mental Illnesses, Physical Illnesses, Education, Year of their birth and other demographic factors.

Overview

Life expectancy is a statistical measure of the average time a human being is expected to live. A typical Regression Machine Learning project leverages historical data to predict insights into the future. This problem statement is aimed at predicting Life Expectancy rate of a country given various features. This problem statement provides a way to predict average life expectancy of people living in a country when various factors such as year, GDP, education, alcohol intake of people in the country, expenditure on healthcare system and some specific disease related deaths that happened in the country are given in a dataset.

In order to predict life expectancy rate of a given country, we will be using Machine Learning algorithms to draw inferences from the given dataset and give an output. For better usability by the customer, we are also going to be creating a UI for the user to interact with using Node-Red.

Purpose

The purpose of this project is that the people from various places can easily predict their life expectancy by providing the inputs asked by the model. This software can be used by all people in the world because the training part of this model contains inputs and predictions of more number of countries.

Economic growth:

Predicting life expectancy would play a vital role in judging the growth and development of the economy.

Across countries, high life expectancy is associated with high income per capita. Increase in life expectancy also leads to an increase in the “manpower” of a country. The knowledge asset of a country increases with the number of individuals in a country.

Population Growth:

Helps the government bodies take appropriate measures to control the population growth and also direct the utilization of the increase in human resources and skillset acquired by people over many years.

Personal growth:

This project would also help an individual assess his/her lifestyle choices and alter them accordingly to lead a longer and healthier life. It would make them more aware of their general health and its improvement or deterioration over time.

Growth in Health Sector:

Based on the factors used to calculate life expectancy of an individual and the outcome, health care will be able to fund and provide better services to those with greater need.

Insurance Companies:

Insurance sector will be able to provide individualized services to people based on the life expectancy outcomes and factors.

2. Literature Survey

There are so many organizations that are making research in the prediction of life expectancy. Many research papers dealing with the creation of this model under many algorithms such as Machine Learning, Deep learning and programming languages such as Python and Java script.

Existing Problem

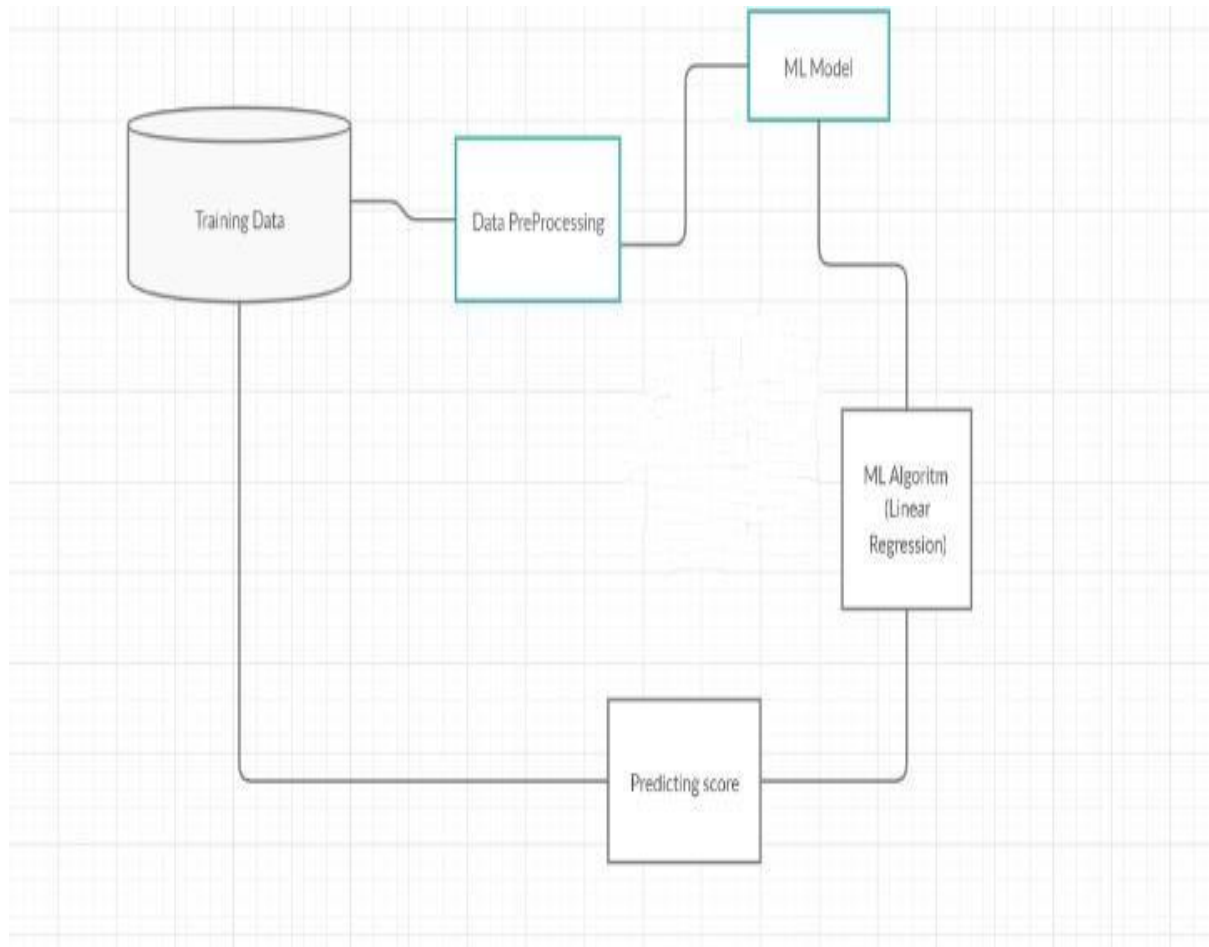
The World Health Organization (WHO) began producing annual life tables for all Member States in 1999. These life tables are a basic input to all WHO estimates of global, regional and country-level patterns and trends in all-cause and cause-specific mortality. After the publication of life tables for years to 2009 in the 2011 edition of World Health Statistics, WHO has shifted to a two year cycle for the updating of life tables for all Member States. Even still the model is not really updated in every fields. WHO applies standard methods to the analysis of Member State data to ensure comparability of estimates across countries. This will inevitably result in differences for some Member States with official estimates for quantities such as life expectancy, where a variety of different projection methods and other methods are used.

Proposed Solution

So many people were expecting to use a model of life expectancy prediction. In order to that, many institutions and companies are leading their team to build that model. In my project, I have proposed a solution to predict the life expectancy using machine learning. Machine Learning is the process of training the computer to think and decide solutions like human. The reason why I have chosen this architecture was only with the help of Machine Learning, deep understanding of the data and an ability to create a model can be done. Design a Regression model to predict life expectancy ratio of a given country based on some features provided such as year, GDP (gross domestic product), education, alcohol intake of people in the country, expenditure on healthcare system and some specific disease related deaths that happened in the country.

3. Theoretical Analysis

Block Diagram



Hardware / Software Designing

1. PROJECT PLANNING AND KICKOFF:
 - a. Understanding the project description and analyze the data and attributes in the given dataset.
 - b. Creating Github account
 - c. Installing Slack and create account with the mail id
 - d. Learning to use Zoho writer.
2. EXPLORE IBM CLOUD PLATFORM:
 - a. Creating IBM cloud account with the mail id
 - b. Creating IBM academic initiative account with the mail id
 - c. Create a Node-Red starter application.
3. EXPLORE IBM WATSON SERVICES:
 - a. Exploring IBM Watson use cases.
 - b. Learning about IBM Watson Machine Learning.
4. INTRODUCTION TO WATSON STUDIO:
 - a. Learning to build own Machine Learning model using IBM Watson.
 - b. Automate the Machine Learning Model
5. PREDICTING LIFE EXPECTANCY WITH PYTHON:
 - a. Collecting Data set from www.kaggle.com
 - b. Creating IBM Watson services
 - c. Create a jupyter notebook and import data from Object storage.
6. PREDICTING LIFE EXPECTANCY WITHOUT PYTHON:
 - a. Created Node-Red model and integrated with Machine Learning model.

4. Experimental Investigation

Life Expectancy Dataset:

The dataset used is a life expectancy dataset released by the World Health Organization.

The data set has the following features:

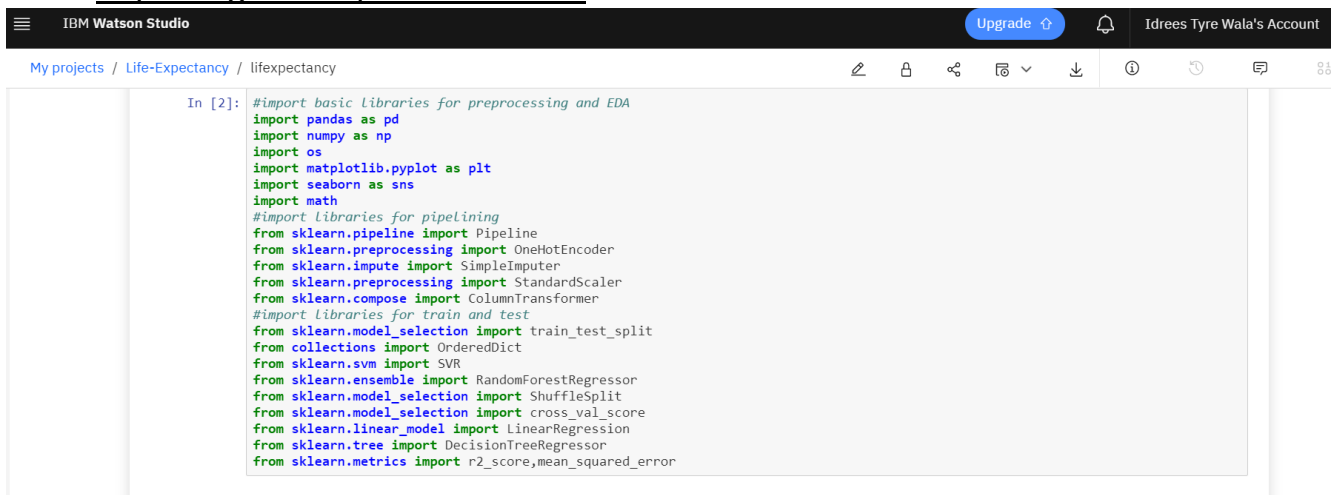
The data is saved as a csv file as LifeExpectancy.csv and it is read and stored in the life data variable. The Year column is dropped as it will not be used in the analysis. The first 5 rows are shown below. The data contains 21 columns and 2938 rows with the header row. The table contains data about:

- Countries
- Status
- Life Expectancy
- Adult Mortality
- Alcohol
- percentage expenditure
- Hepatitis B
- Measles
- BMI
- under-five deaths
- Polio
- Total expenditure
- Diphtheria
- HIV/AIDS
- GDP
- Population
- thinness 1-19 years
- thinness 5-9 years
- Income composition of resources
- Schooling

Preprocessing and cleaning the datasets:

- Before the data can be imported using the machine learning libraries and can be trained, the data needs to be cleaned and pre-processed.
- All the null values in the data set need to be either set to 0, deleted or set equal to the mean value.
- In the cleaning process, I have set the null values as 0 for the ease of calculation and maintaining the accuracy of the model.

Importing the required libraries:



The screenshot shows the IBM Watson Studio interface. At the top, there's a header with 'IBM Watson Studio', an 'Upgrade' button, and a user account 'Idrees Tyre Wala's Account'. Below the header, the breadcrumb navigation shows 'My projects / Life-Expectancy / lifexpectancy'. The main area displays a Jupyter Notebook cell with the following code:

```
In [2]: #import basic libraries for preprocessing and EDA
import pandas as pd
import numpy as np
import os
import matplotlib.pyplot as plt
import seaborn as sns
import math

#import Libraries for pipelining
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import OneHotEncoder
from sklearn.impute import SimpleImputer
from sklearn.preprocessing import StandardScaler
from sklearn.compose import ColumnTransformer

#import Libraries for train and test
from sklearn.model_selection import train_test_split
from collections import OrderedDict
from sklearn.svm import SVR
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import ShuffleSplit
from sklearn.model_selection import cross_val_score
from sklearn.linear_model import LinearRegression
from sklearn.tree import DecisionTreeRegressor
from sklearn.metrics import r2_score, mean_squared_error
```

Loading the packages:

The following packages have been imported NumPy, Pandas, Matplotlib, SciPy, and Seaborn. Sklearn is the most widely used package for the machine learning process. The following subpackages have been used:

1. train_test_split
2. linear_model
3. model selection
4. metrics
5. tree
6. ensemble
7. pre-processing

Training the regression model:

IBM Watson Studio

Upgrade

Idrees Tyre Wala's Account

My projects / Life-Expectancy / lifexpectancy

Developed	78.83
Developing	66.19

TRAINING AND TESTING

```
In [33]: X = df[['Year', 'Status', 'Country',
'winsorized_Adult_Mortality', 'winsorized_Infant_Deaths',
'winsorized_Alcohol', 'winsorized_Percentage_Exp',
'winsorized_HepatitisB', 'winsorized_Measles', 'BMI', 'winsorized_Under_Five_Deaths',
'winsorized_Polio', 'winsorized_Tot_Exp', 'winsorized_Diphtheria',
'winsorized_HIV', 'winsorized_GDP', 'winsorized_Population',
'winsorized_thinness_10_19_years', 'winsorized_thinness_5_9_years',
'winsorized_Income_Comp_OF_Resources', 'winsorized_Schooling']]
y = df['winsorized_Life_Expectancy']
X.head()
```

Out[33]:

	Year	Status	Country	winsorized_Adult_Mortality	winsorized_Infant_Deaths	winsorized_Alcohol	winsorized_Percentage_Exp	winsoriz
0	2015	Developing	Afghanistan	263.0	61	0.01	71.279624	65.0
1	2014	Developing	Afghanistan	271.0	61	0.01	73.523582	62.0
2	2013	Developing	Afghanistan	268.0	61	0.01	73.219243	64.0
3	2012	Developing	Afghanistan	272.0	61	0.01	78.184215	67.0
4	2011	Developing	Afghanistan	275.0	61	0.01	7.097109	68.0

5 rows × 21 columns

```
In [34]: y.head()
```

Out[34]:

```
0    65.0
1    59.9
2    59.9
3    59.5
4    59.2
Name: winsorized_Life_Expectancy, dtype: float64
```

```
In [35]: X_train, X_test, Y_train, Y_test = train_test_split(X, y, test_size = 0.2, random_state = 42)
```

```
In [36]: X_train.head()
```

Out[36]:

	Year	Status	Country	winsorized_Adult_Mortality	winsorized_Infant_Deaths	winsorized_Alcohol	winsorized_Percentage_Exp	wins
1777	2002	Developing	Mozambique	416.0	61	2.16	40.825971	76.0
2631	2001	Developing	Togo	345.0	14	0.95	2.048575	24.0
500	2011	Developing	Canada	68.0	2	8.20	971.928038	18.0
1462	2011	Developing	Lebanon	93.0	1	1.57	835.062683	81.0
1736	2011	Developing	Montenegro	113.0	0	6.56	666.737437	91.0

5 rows × 21 columns

```
In [37]: #IDENTIFY THE NUMERIC VALUES FOR COLUMNTRANSFORM
numeric_features = ['Year',
'winsorized_Adult_Mortality', 'winsorized_Infant_Deaths',
'winsorized_Alcohol', 'winsorized_Percentage_Exp',
'winsorized_HepatitisB', 'winsorized_Measles', 'BMI', 'winsorized_Under_Five_Deaths',
'winsorized_Polio', 'winsorized_Tot_Exp', 'winsorized_Diphtheria',
'winsorized_HIV', 'winsorized_GDP', 'winsorized_Population',
'winsorized_thinness_10_19_years', 'winsorized_thinness_5_9_years',
'winsorized_Income_Comp_OF_Resources', 'winsorized_Schooling']
```

Predictions from our model:

```
Out[42]: (array([0.81873949, 0.95340768]),
          (('Linear Regression', 0.8187394918185045),
           ('Random Forest Regressor', 0.9534076876019453)))
```

```
In [42]: RF = Pipeline([
          ( 'preprocessor', preprocessor),
          ( 'RF', RandomForestRegressor())
        ])
```

```
In [43]: RF.fit(X_train,Y_train)
```

```
/opt/conda/envs/Python36/lib/python3.6/site-packages/sklearn/ensemble/forest.py:246: FutureWarning: The default value of n_estimators will change from 10 in version 0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
```

```
Out[43]: Pipeline(memory=None,
                  steps=[('preprocessor', ColumnTransformer(n_jobs=None, remainder='drop', sparse_threshold=0.3,
                  transformer_weights=None,
                  transformers=[('cat', Pipeline(memory=None,
                  steps=[('onehot', OneHotEncoder(categorical_features=None, categories=None,
                  dtype=<class 'numpy.float64'>...ators=10, n_jobs=None,
                  oob_score=False, random_state=None, verbose=0, warm_start=False))]))])
```

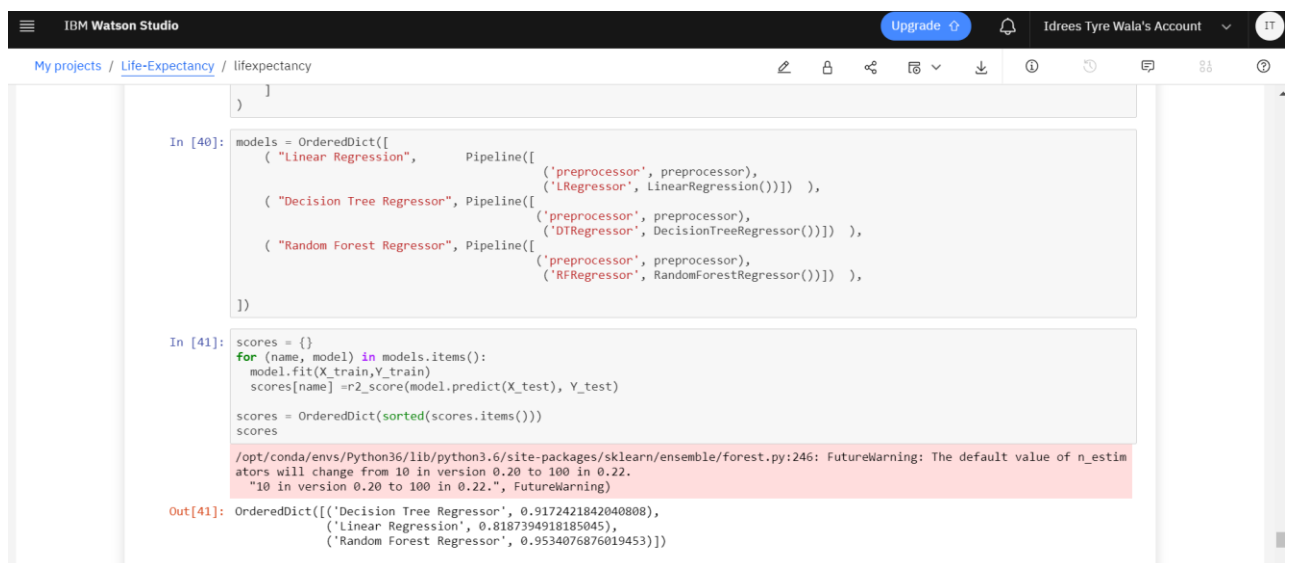
```
In [44]: predict= RF.predict(X_test)
```

```
In [45]: r2_score(predict,Y_test)
```

```
Out[45]: 0.9549919687358972
```

MODEL BUILDING AND DEPLOYMENT

Linear regression with polynomial functions:



The screenshot shows the IBM Watson Studio interface. The top bar includes the IBM Watson Studio logo, an 'Upgrade' button, a notification bell, and the user's account name 'Idrees Tyre Wala's Account'. The breadcrumb navigation shows 'My projects / Life-Expectancy / lifeexpectancy'. The notebook contains two code cells. The first cell, labeled 'In [40]:', defines three models in an OrderedDict: 'Linear Regression' (using LinearRegression), 'Decision Tree Regressor' (using DecisionTreeRegressor), and 'Random Forest Regressor' (using RandomForestRegressor). Each model is wrapped in a Pipeline with a preprocessor. The second cell, labeled 'In [41]:', calculates the R-squared score for each model on a test set. It iterates through the models, fits them on training data, and calculates the r2_score on test data. The output, labeled 'Out[41]:', shows the R-squared scores for each model: Decision Tree Regressor (0.9172421842040808), Linear Regression (0.8187394918185045), and Random Forest Regressor (0.9534076876019453). A FutureWarning message is also visible, indicating that the default value of n_estimators will change from 10 to 100 in version 0.22.

```
In [40]: models = OrderedDict([
    ( "Linear Regression", Pipeline([
        ('preprocessor', preprocessor),
        ('LRegressor', LinearRegression()) ] ) ),
    ( "Decision Tree Regressor", Pipeline([
        ('preprocessor', preprocessor),
        ('DTRRegressor', DecisionTreeRegressor()) ] ) ),
    ( "Random Forest Regressor", Pipeline([
        ('preprocessor', preprocessor),
        ('RFRegressor', RandomForestRegressor()) ] ) ),
])

In [41]: scores = {}
for (name, model) in models.items():
    model.fit(X_train, Y_train)
    scores[name] = r2_score(model.predict(X_test), Y_test)

scores = OrderedDict(sorted(scores.items()))

/opt/conda/envs/Python36/lib/python3.6/site-packages/sklearn/ensemble/forest.py:246: FutureWarning: The default value of n_estimators will change from 10 in version 0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)

Out[41]: OrderedDict([('Decision Tree Regressor', 0.9172421842040808),
 ('Linear Regression', 0.8187394918185045),
 ('Random Forest Regressor', 0.9534076876019453)])
```

Three models have been created. The Algorithms have been used to test if they can provide good prediction with fewer errors while predicting the life expectancy for new data. The Model Algorithms used are:

- Decision Tree Regression
- Linear Regression
- Random Forest Regression

On Comparing Both the Models, we came to this conclusion that Random Forest Model is giving us less error and best Prediction score in compare to Linear Regression Model and Decision Tree Regression.

Deployment of Model:

```
IBM Watson Studio Upgrade Idrees Tyre Wala's Account IT

My projects / Life-Expectancy / lifexpectancy

'url': 'https://eu-gb.ml.cloud.ibm.com/v3/ml_instances/340bacf2-d172-42c7-973f-b81108144c5c/published_models/18d336ef-a01f-4987-a41d-5f37aa453ae6/deployments'},
'evaluation_metrics_url': 'https://eu-gb.ml.cloud.ibm.com/v3/ml_instances/340bacf2-d172-42c7-973f-b81108144c5c/published_models/18d336ef-a01f-4987-a41d-5f37aa453ae6/evaluation_metrics'}}

In [77]: published_model_uid = client.repository.get_model_uid(model_artifact)
published_model_uid

Out[77]: '18d336ef-a01f-4987-a41d-5f37aa453ae6'

In [78]: deployment = client.deployments.create(published_model_uid, name="Life Expectancy")
scoring_endpoint = client.deployments.get_scoring_url(deployment)
scoring_endpoint

#####

Synchronous deployment creation for uid: '18d336ef-a01f-4987-a41d-5f37aa453ae6' started

#####

INITIALIZING
DEPLOY_SUCCESS

-----

Successfully finished deployment creation, deployment_uid='ab13d335-5d0d-4cb5-8800-f0201fd26c46'

-----

Out[78]: 'https://eu-gb.ml.cloud.ibm.com/v3/ml_instances/340bacf2-d172-42c7-973f-b81108144c5c/deployments/ab13d335-5d0d-4cb5-8800-f0201fd26c46/online'

In [79]: #GET SCORING END-POINT URL
scoring_endpoint = client.deployments.get_scoring_url(deployment)
print(scoring_endpoint)

https://eu-gb.ml.cloud.ibm.com/v3/ml_instances/340bacf2-d172-42c7-973f-b81108144c5c/deployments/ab13d335-5d0d-4cb5-8800-f0201fd26c46/online

In [80]: client.deployments.list()

-----

GUID NAME TYPE STATE CREATED FRAMEWORK ART
INFANT TYPE
ab13d335-5d0d-4cb5-8800-f0201fd26c46 Life Expectancy online DEPLOYING SUCCESS 2020-05-17T14:17:00.000Z Python 3.7.0 64-bit Linux x86_64
```

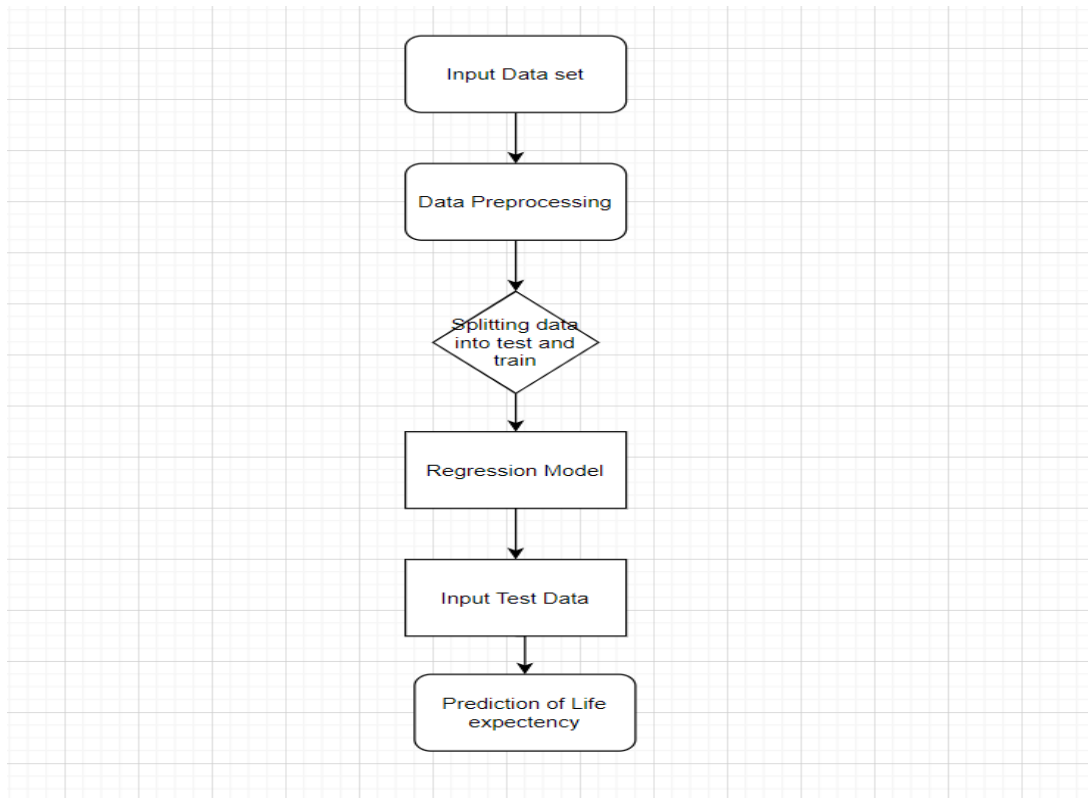
Testing the Deployment:

```
-----

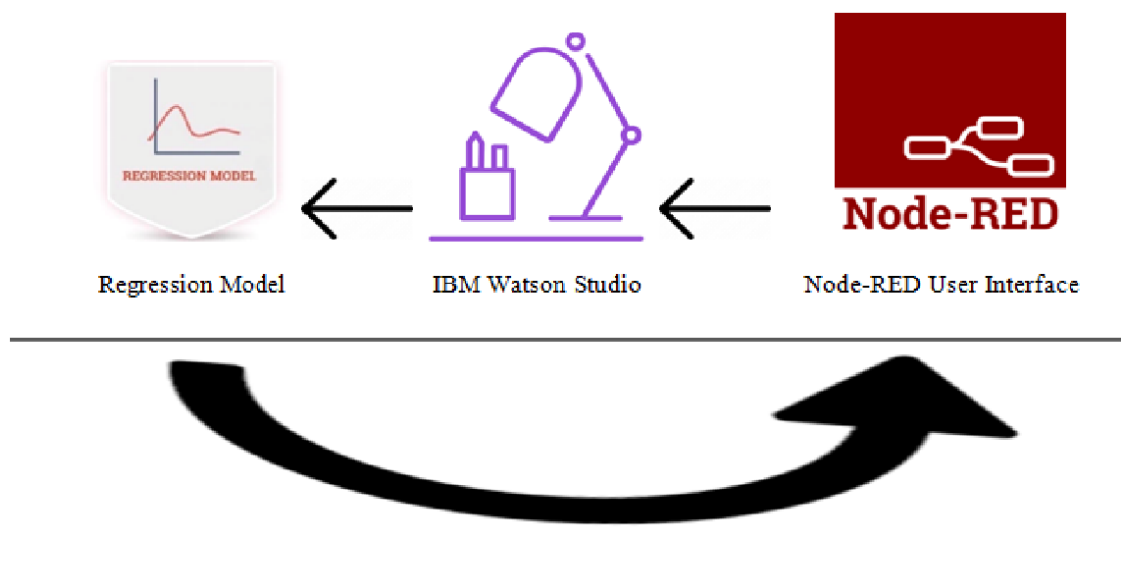
In [81]: #TEST THE DEPLOYMENT
scoring_payload = {"fields": ['BMI', 'HIV/AIDS', 'thinness 1-19 years', 'thinness 5-9 years',
'Adult Mortality', 'Alcohol', 'Country', 'Diphtheria', 'GDP',
'Hepatitis B', 'Income composition of resources', 'Measles', 'Polio',
'Population', 'Schooling', 'Status', 'Total expenditure', 'Year',
'infant deaths', 'percentage expenditure', 'under-five deaths'], "values": [[19.1, 0.1, 17.2, 17.3, 263, 0.01, 'Afghanistan', 65,
584.25, 65, 0.47, 1154, 6, 33736494, 10, 'Developing', 8.16, 2015, 62, 71.27, 83]]}
predictions = client.deployments.score(scoring_endpoint, scoring_payload)
print(predictions)

{'fields': ['prediction'], 'values': [[72.61]]}
```

5. Flowchart



UI USING THE NODE RED



To integrate the ML model with the UI, we would be using the Node Red functionality provided by the IBM Watson Studio.

To design the UI, we need to import the flow of the UI.

Once, we have setup the flow, we need to integrate the ML model with it. To integrate the ML Model with it we need to access the endpoint URL of our ML Model.

Components of the flow are:

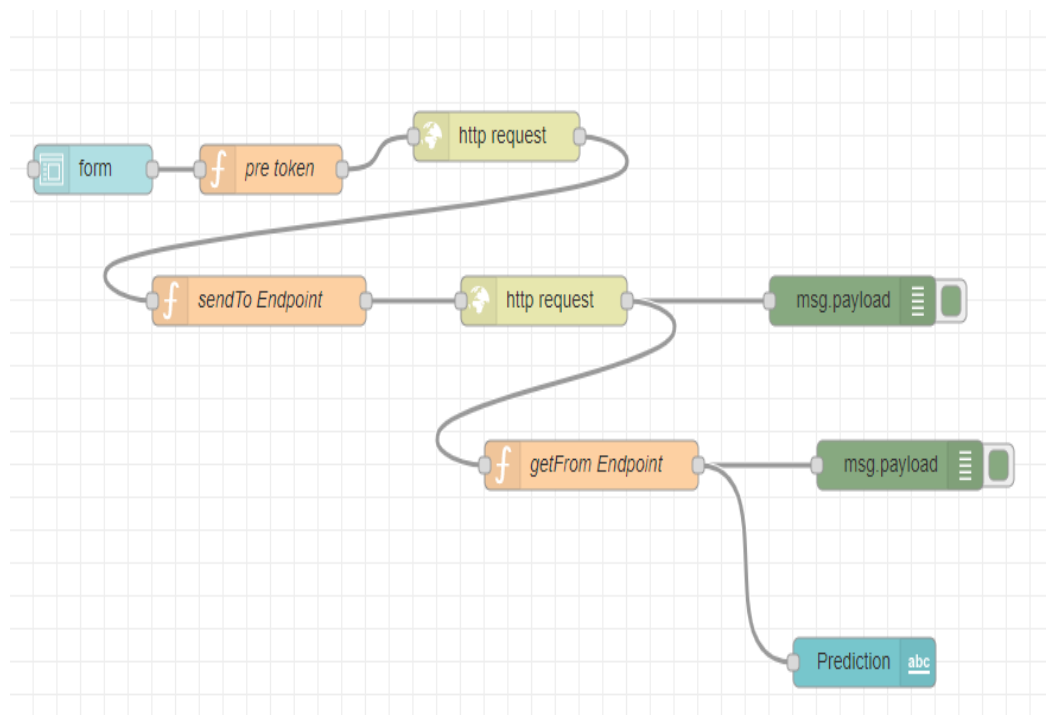
Form: The form contains all the elements of the UI. All the labels are associated with a variable.

Http requests: To setup the flow, we need two http requests.

The first http request requires a token to connect to the machine learning service of the Watson studio.

The second http request helps us in integrating the model using the endpoint URL.

Once the flow has been setup, we deploy the model.



6. Result

Web based UI was developed by integrating all the services using NODE-RED.

URL for UI Dashboard: https://node-red-eidgp.eu-gb.mybluemix.net/ui/#!/0?socketid=WRLd_oXswAGxMG7qAAAE

URL for Notebook: <https://github.com/SmartPracticeschool/IISPS-INT-2058-Predicting-Life-Expectancy-using-Machine-Learning/blob/master/lifexpectancy.ipynb>

While giving the inputs for the country Albaniaia in the year 2017, the life expectancy value 73.35 has been predicted.

The screenshot shows a web browser window with a blue header bar labeled "Home Page". The main content area is titled "Life Expectancy Prediction" in blue text. Below the title, there is a form with several input fields, each with a label and a value. The "Prediction" field shows the value "73.35". The other fields are labeled "Country *", "Year *", "Status *", "BMI *", "Adult Mortality *", "Infant Deaths *", "Alcohol *", "Percentage Expenditure *", and "Hepatitis B *". The values entered in these fields are "Albania", "2015", "Developing", "58", "74", "0", "4.6", "435", and "3" respectively. The form is styled with a light blue background and blue borders.

Life Expectancy Prediction	
Prediction	73.35
Country *	Albania
Year *	2015
Status *	Developing
BMI *	58
Adult Mortality *	74
Infant Deaths *	0
Alcohol *	4.6
Percentage Expenditure *	435
Hepatitis B *	3

Page

HIV/AIDS *

45

GDP *

3543

Population *

54

Thinness 10-19 years *

345

Thinness 5-9 years *

354

Income Composition of Resources *

54

Schooling *

354

Measles *

45

PREDICT

CANCEL

7. Advantages & Disadvantages

Advantages:

One of the biggest advantages of embedding machine learning algorithms is their ability to improve over time. Machine learning technology typically improves efficiency and accuracy thanks to the ever-increasing amounts of data that are processed.

The application learns the patterns and trends hidden within the data without human intervention which makes predicting much simpler and easier. The more data is fed to the algorithm, the higher the accuracy of the algorithm is. It is also the key component in technologies for automation.

Using Node-Red also simplifies the effort put into creating the front-end. The programmer doesn't need extensive knowledge on HTML and JavaScript. It also makes the integration between Machine learning model and the UI much easier.

Disadvantages:

Using machine learning interface comes with its own problems. Since the whole point of it is minimize human involvement, it also makes error detection and fixing much more problematic. It takes a lot of time to identify the root cause for the problem.

Machine learning can also be very time-consuming. When the size of the data fed to the machine learning is very large, the computational cost and the time taken to train the model on the data increases drastically. This can increase the cost of resources required to implement the application on a large scale.

At the same time, Node-Red does not give many features to customize our UI.

8. Applications

- Personalized Life Expectancy: Individuals can predict their own life expectancy by inputting values in the corresponding fields. This could help make people more aware of their general health, and its improvement or deterioration over time. This may motivate them to make healthier lifestyle choices.
- Government: It could help the government bodies take appropriate measures to control the population growth and also direct the utilization of the increase in human resources and skillset acquired by people over many years. Across countries, high life expectancy is associated with high income per capita. Increase in life expectancy also leads to an increase in the “manpower” of a country. The knowledge asset of a country increases with the number of individuals in a country.
- Health Sector: Based on the factors used to calculate life expectancy of an individual and the outcome, health care will be able to fund and provide better services to those with greater need.
- Insurance Companies: Insurance sector will be able to provide individualized services to people based on the life expectancy outcomes and factors.

9. Conclusion

- The end product is a webpage created and deployed on node-red app of IBM cloud. The backend of webpage is a linear regression model created and deployed on Watson Studio using machine learning service.
- This model can be used to predict the life expectancy of people in different places.
- This model contains various factors such as Country, Year, Status, Life Expectancy, Adult Mortality, Infant Deaths, Alcohol, Percentage Expenditure, Hepatitis B, Measles, BMI, Under-Five Deaths, Polio, Total Expenditure, Diphtheria, HIV/AIDS, GDP, Population, Thinness 1-19 Years, Thinness 5-9 Years, Income Composition Of Resources, Schooling.
- With the help of all these input values, the model will predict the life expectancy of such people.
- The accuracy level of prediction in my model is more than 95%.
- From the help of this model, the life expectancies of more than 190 countries can be detected.

10. Future Scope

For future use, one can integrate the life expectancy prediction with providing suggestions and medications to the individual using the application. This will help predict as well as increase the individual's life expectancy.

The scalability and flexibility of the application can also be improved with advancement in technology and availability of new and improved resources. Also, with the growth in Artificial Neural networks and Deep learning, one can integrate that with our existing application. With the help of Convolutional Neural networks and Computer vision, we can also try to take into account the physical health and appearance of a person.

Mental health can also be taken into account while predicting life expectancy with the help of sentiment analysis systems as well.

11. Bibliography

1. Node-RED Starter Application :

<https://developer.ibm.com/tutorials/how-to-create-a-node-red-starter-application/>

2. Watson Studio Cloud :

<https://bookdown.org/caoying4work/watsonstudio-workshop/jn.html>

3. Dataset Reference: -

<https://www.kaggle.com/kumaraia/rshi/life-expectancy-who>

4. IBM Cloud Services :

<https://www.youtube.com/watch?v=DBRGIAHdj48&list=PLzpeuWUENMK2PYtasCaKK4bZjaYzhW23L>

5. Import the Dataset into Jupyter Notebook :

<https://www.youtube.com/watch?v=Jtej3Y6uUng>

